

## Κεφάλαιο 1

### ΚΑΤΑΝΟΗΣΗ ΚΑΘΟΡΙΣΜΟΣ ΚΑΙ ΔΟΜΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

#### Ερωτήσεις

##### 1. Τι καλείται κατανόηση προβλήματος;

Κατανόηση του προβλήματος είναι η εργασία κατά την οποία μελετάμε προσεκτικά το περιεχόμενο του προβλήματος ώστε να **διακρίνουμε το συσχετισμό των δεδομένων με τα ζητούμενά του**, αποσαφηνίζοντας ακόμα και παρερμηνίες οι οποίες προκύπτουν από τη διατύπωσή του.

##### 2. Πότε επιβάλλεται να χωρίσουμε ένα πρόβλημα σε άλλα απλούστερα;

Επιβάλλεται να χωρίσουμε ένα πρόβλημα σε άλλα απλούστερα, όταν είναι **μεγάλο** ή όταν απαιτεί **πολύπλοκο τρόπο επίλυσης**.

##### 3. Τι καλείται δομή προβλήματος;

Δομή προβλήματος ονομάζονται τα **απλούστερα προβλήματα** από τα οποία αποτελείται καθώς και ο **τρόπος που συνδέονται**, έτσι ώστε η επίλυσή τους να επιλύει το όλο πρόβλημα [Σημ. υπάρχει κακοδιατύπωση για τον όρο «δομή προβλήματος», τόσο στη σελ. 5 όσο και στη σελίδα 6 στην Περίληψη].

##### 4. Τι καλείται καθορισμός προβλήματος;

Καθορισμός προβλήματος είναι η εργασία η οποία **ακολουθεί** την ανάλυση του προβλήματος σε απλούστερα. Κατά τον καθορισμό του προβλήματος γίνεται ο **προσδιορισμός των δεδομένων** του προβλήματος καθώς και των **ζητούμενων**. [Σημ. Στην Περίληψη εισάγεται η έννοια «Καθορισμός απαιτήσεων προβλήματος» η οποία εξετάζεται στο Κεφάλαιο 2].

##### 5. Να συμπληρώσετε τα κενά με τη λέξη που λείπει:

α. Εάν πρόβλημα το χωρίζουμε σε άλλα απλούστερα όταν είναι **μεγάλο** ή όταν έχει **πολύπλοκο τρόπο επίλυσης**.

β. Η επίλυση ενός προβλήματος προϋποθέτει την **κατανόησή** του.

γ. Τα προβλήματα δεν παραμένουν **στατικά** με το πέρασμα του χρόνου αλλά τα στοιχεία τους μεταβάλλονται **δυναμικά**.

Δ. Μετά την κατανόηση του προβλήματος και διάκρισή του σε απλούστερα, δεν απομένει παρά να προσδιορίσουμε τα **δεδομένα** του προβλήματος καθώς και τα **ζητούμενα**.

##### 6. Να σημειώσετε με Σωστό – Λάθος τις παρακάτω προτάσεις:

α. Η διοργάνωση μιας εκδρομής αποτελεί πρόβλημα που απαιτεί επίλυση. (Σ)

β. Η σαφήνεια στη διατύπωση ενός προβλήματος είναι δεδομένη. (Λ)

γ. Κάθε πρόβλημα μπορεί να επιλυθεί με τη χρήση υπολογιστή (Λ)

## Κεφάλαιο 2

### ΚΑΘΟΡΙΣΜΟΣ ΑΠΑΙΤΗΣΩΝ

#### Ερωτήσεις

##### 1. Πότε εξυπηρετεί η λύση ενός προβλήματος με τη βοήθεια υπολογιστή3;

Η χρήση υπολογιστή για την λύση ενός προβλήματος εξυπηρετεί όταν:

- Έχουμε μεγάλο πλήθος δεδομένων να επεξεργαστούμε.
- Τα αποτελέσματα που παράγονται είναι πολλά.
- Η μέθοδος επίλυσης είναι πολύπλοκη για τον άνθρωπο.
- Η μέθοδος επίλυσης επαναλαμβάνεται πολλές φορές.

##### 2. Τι ονομάζουμε δεδομένα ενός προβλήματος;

**Δεδομένα** ενός προβλήματος ονομάζεται το **σύνολο** εκείνων των **στοιχείων**, τα οποία είναι **απαραίτητα για την επίλυση του προβλήματος**.

##### 3. Τι καλείται καθορισμός απαιτήσεων ενός προβλήματος4;

**Καθορισμός των απαιτήσεων** του προβλήματος είναι η εργασία η οποία περιλαμβάνει τη σαφή **καταγραφή των δεδομένων**, την αναλυτική **διατύπωση του προβλήματος**, τη λεπτομερειακή **περιγραφή των ζητούμενων** και τη **διερεύνηση των αποτελεσμάτων**.

##### 4. Ποιοι είναι οι παράγοντες που επηρεάζουν τον καθορισμό των απαιτήσεων για την επίλυση ενός προβλήματος;

Οι **παράγοντες** που επηρεάζουν τον καθορισμό των απαιτήσεων ενός προβλήματος είναι:

- Η **επιλογή δεδομένων** που έχουν άμεση σχέση με το πρόβλημά μας.
- Ο έλεγχος **ορθότητας των δεδομένων** του προβλήματος.
- Ο έλεγχος **πληρότητας των δεδομένων** σε σχέση με το ζητούμενο αποτέλεσμα του προβλήματος.
- Ο καθορισμός των **ζητούμενων** του προβλήματος.

##### 5. Γιατί η καταχώριση των δεδομένων ενός προβλήματος στον υπολογιστή δεν αποτελεί μέρος του καθορισμού των απαιτήσεων ενός προβλήματος;

Η καταχώριση των δεδομένων του προβλήματος (σ.σ. αν γίνεται σωστά ή όχι) δεν έχει σχέση με τον ορισμό (καταγραφή) των δεδομένων που είναι εργασία η οποία γίνεται κατά τον καθορισμό των απαιτήσεων του

προβλήματος. [Για παράδειγμα, στο πρόβλημα της εξαγωγής του ΜΟ των μαθημάτων ενός μαθητή Γυμνασίου, ΟΡΙΖΟΥΜΕ ότι οι βαθμοί πρέπει να βρίσκονται στην κλίμακα από 1 μέχρι 20. Αυτό είναι διαφορετικό πράγμα από τη σωστή καταχώρηση της βαθμολογίας].

#### 6. Να συμπληρώσετε τα κενά με τη λέξη που λείπει:

α. Εκτός από την κατανόηση του προβλήματος εξίσου σημαντικός είναι και ο **ακριβής προσδιορισμός** των δεδομένων.

β. Σημαντικό είναι να ελέγχουμε την **ορθότητα** των δεδομένων σε σχέση με το επιδιωκόμενο αποτέλεσμα, καθώς και την **πληρότητα** των δεδομένων σε σχέση με τα απαραίτητα για την επίλυση του προβλήματος.

3 Δεν αναφέρεται κάτι πάνω σε αυτό στο Κεφάλαιο 2 του βιβλίου. 4 Στην Περίληψη του Κεφαλαίου 2 περιγράφεται ο «καθορισμός απαιτήσεων του προβλήματος» (η ίδια έννοια έχει αναφερθεί και στην Περίληψη του Κεφαλαίου 1) με τρόπο παρόμοιο με εκείνον της έννοιας «καθορισμός προβλήματος» στη λεζάντα της σ.6. Πρόκειται προφανώς(;) για την ίδια έννοια.

#### 7. Να σημειώσετε με Σ- Λ τις παρακάτω προτάσεις:

α. Ο έλεγχος της ορθότητας των δεδομένων αποτελεί μέρος της κατανόησης τους προβλήματος (Λ – βλέπε Περίληψη Κεφαλαίου 1 – στάδια επίλυσης προβλήματος).

β. Οι πληροφορίες που προέρχονται από την επεξεργασία των δεδομένων ενός προβλήματος μπορούν να χρησιμοποιηθούν ως δεδομένα για το ίδιο πρόβλημα (Λ –βλέπε σελ. 11 σειρές 7-9).

γ. Ο έλεγχος της ορθής καταχώρησης δεδομένων ενός προβλήματος στον υπολογιστή, αποτελεί μέρος του καθορισμού των απαιτήσεων του προβλήματος; (Λ – βλέπε σελ. 11- 3η παράγραφος).

### Κεφάλαιο 3

#### ΑΛΓΟΡΙΘΜΟΙ - ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

##### Ερωτήσεις

#### 1. Τι καλείται αλγόριθμος;

Αλγόριθμος καλείται η διαδοχική σειρά πεπερασμένου πλήθους λογικών βημάτων τα οποία εκτελούνται σε πεπερασμένο χρόνο και τα οποία απαιτούνται για την επίλυση ενός προβλήματος.

#### 2. Ποια είναι τα χαρακτηριστικά ενός καλού αλγορίθμου;

Τα χαρακτηριστικά ενός καλού αλγορίθμου είναι:

- Να περιγράφει με **ακρίβεια τα δεδομένα** που ο ίδιος θα χρησιμοποιήσει.
- Κάθε βήμα του να είναι τελείως **καθορισμένο** και να φανερώνει τι ακριβώς κάνει.
- Να δίνει με **ακρίβεια τα αποτελέσματα** που παράγονται από το κάθε βήμα του.
- Να **ολοκληρώνει πάντοτε σωστά τη σκέψη επίλυσης** του προβλήματος ύστερα από έναν **πεπερασμένο** αριθμό βημάτων, που το καθένα εκτελείται σε **πεπερασμένο χρόνο**.

#### 3. Να συμπληρώσετε τα κενά με τις λέξεις που λείπουν.

α. Απαραίτητο και σπουδαίο είναι η **δημιουργία αλγορίθμου** για την επίλυση ενός προβλήματος.

β. Η διαδοχική σειρά **πεπερασμένου πλήθους** λογικών βημάτων που απαιτούνται για την επίλυση ενός προβλήματος καλείται αλγόριθμος.

### Κεφάλαιο 4

#### ΑΝΑΠΤΥΞΗ ΑΛΓΟΡΙΘΜΩΝ

##### Ερωτήσεις

#### 1. Τι καλείται ψευδοκώδικας;

**Ψευδοκώδικας** καλείται η καταγραφή του αλγορίθμου με απλές **λεκτικές** προστακτικές **προτάσεις** και **σύμβολα** τα οποία προέρχονται από διαδοσμένες γλώσσες προγραμματισμού.

#### 2. Τι είναι το λογικό διάγραμμα;

**Λογικό διάγραμμα** είναι η μέθοδος αναπαράστασης του αλγορίθμου με τη χρήση απλών **γεωμετρικών σχημάτων** τα οποία υποστηρίζονται με **απλές λέξεις** για την αναπαράσταση συγκεκριμένων λειτουργιών του αλγορίθμου.

#### 3. Για ποιο λόγο είναι απαραίτητη η τυποποίηση του αλγορίθμου;

Η τυποποίηση του αλγορίθμου επίλυσης ενός προβλήματος είναι απαραίτητη ώστε να είναι δυνατόν να είναι **κατανοητός και από άλλους**, οι οποίοι θα τον μελετήσουν, αλλά και από τον ίδιο **δημιουργό του** όταν μετά από κάποιο καιρό θελήσει να θυμηθεί ή να τροποποιήσει το αλγόριθμο, τον οποίο κατασκεύασε.

#### 4. Ποιες είναι οι βασικές αλγοριθμικές δομές;

Οι βασικές αλγοριθμικές δομές είναι η ακολουθία, η επιλογή και η επανάληψη.

#### 5. Ποια είναι η δομή της ακολουθίας;

Στην ακολουθία, οι εντολές οι οποίες βρίσκονται σε διαδοχή, εκτελούνται με τη σειρά με την οποία είναι γραμμένες.

**6. Ποια είναι η δομή της επιλογής;**

Με τη **δομή επιλογής** παρέχεται η δυνατότητα εκτέλεσης μιας ή περισσότερων εντολών **ανάλογα με το αποτέλεσμα ελέγχου μιας συνθήκης**.

**8. Να συμπληρώσετε τα κενά με τη λέξη που λείπει:**

- α. Είναι απαραίτητο να γίνει μια **τυποποίηση** του τρόπου που θα αναπαρίστανται οι αλγόριθμοι.
- β. Ο ψευδοκώδικας διαδόθηκε περισσότερο, όταν και οι γλώσσες προγραμματισμού έγιναν **απλούστερες** και απέκτησαν έναν τρόπο έκφρασης, πολύ κοντά στον **καθημερινό** τρόπο επικοινωνίας μας.
- γ. Με την εντολή **Επανάλαβε ... μέχρι** έχουμε εκτέλεση τουλάχιστον μία φορά, των εντολών που περιέχει, έστω και αν η συνθήκη είναι αληθής.

**9. Να σημειώσετε με Σ-Λ τις παρακάτω προτάσεις:**

- α. Η καταγραφή του αλγορίθμου είναι απαραίτητη βοήθεια για την επίλυση ενός προβλήματος. (Σ-προφανώς).
- β. Ο ψευδοκώδικας χρησιμοποιεί σχήματα για την αναπαράσταση του αλγορίθμου (Λ).
- γ. Στη δομή επανάληψης **επανάλαβε εφόσον** εκτελούνται εκτελούνται τουλάχιστον μια φορά οι εντολές που περιέχονται. (Λ)

## Κεφάλαιο 5

### ΣΤΟΙΧΕΙΑ ΔΟΜΗΜΗΜΕΝΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ Ερωτήσεις

**1. Τι καλείται τμηματικός προγραμματισμός;**

Τμηματικός προγραμματισμός ονομάζεται η εργασία που ακολουθεί την ιεραρχική σχεδίαση και είναι η υλοποίηση του προγράμματος μέσα από ανεξάρτητες λογικές ενότητες – τμήματα. Τα μεμονωμένα τμήματα πρέπει να είναι όσο το δυνατόν μικρότερα ώστε να είναι εύκολο να διορθωθούν.

**2. Τι καλείται ιεραρχικός προγραμματισμός;**

**Ιεραρχικός προγραμματισμός** καλείται η τεχνική σχεδίασης αλγορίθμου κατά την οποία το πρόβλημα **διασπάται** σε μια σειρά από **απλούστερα προβλήματα**, τα οποία όταν επιλυθούν, οδηγούν στην επίλυση του αρχικού προβλήματος. Η σχεδίαση του προγράμματος πρέπει να προχωρά από **πάνω προς τα κάτω**. Στα ανώτερα επίπεδα ξεκινάμε από τα **γενικά** και προχωρούμε σε κατώτερα επίπεδα **προσθέτοντας λεπτομέρειες**. Η ανάλυση σε κατώτερα επίπεδα σταματά όταν η επεξεργασία περιέχει τόσες λεπτομέρειες ώστε να μπορεί να κωδικοποιηθεί σε μια γλώσσα προγραμματισμού.

**3. Τι καλείται δομημένος προγραμματισμός;**

**Δομημένος προγραμματισμός** καλείται η τεχνική σχεδίασης αλγορίθμου και του αντίστοιχου προγράμματος, η οποία χρησιμοποιεί τις αρχές του **ιεραρχικού** και του **τμηματικού προγραμματισμού**. Στον δομημένο προγραμματισμό ο **αλγόριθμος** αποτελείται από **ανεξάρτητα τμήματα** με βάση ένα προκαθορισμένο σχέδιο. Έτσι, με τον ίδιο τρόπο θα αναπτυχθεί αμέσως μετά και η κωδικοποίηση του αλγορίθμου σε γλώσσα προγραμματισμού δημιουργώντας τελικά το πρόγραμμα. Ο δομημένος προγραμματισμός χρησιμοποιεί μόνο τις βασικές αλγοριθμικές δομές της **ακολουθίας**, της **επιλογής** και της **επανάληψης** ενώ αποφεύγει τη χρήση της εντολής GO TO. Ένα δομημένο πρόγραμμα διαβάζεται και συντηρείται εύκολα.

**4. Ποια είναι τα πλεονεκτήματα του δομημένου προγραμματισμού;**

Τα κύρια πλεονεκτήματα του δομημένου προγραμματισμού είναι:

- Διευκόλυνση στην ανάπτυξη αλγορίθμων **κατά τμήματα**.
- Ευκολία και ταχύτητα στην **κωδικοποίηση**.
- Καλύτερη **ποιότητα** προγραμμάτων.
- Ευκολία στις **διορθώσεις** και τη συντήρηση.
- **Τεκμηρίωση** που περιέχεται σχεδόν εξ ολοκλήρου στο ίδιο το πρόγραμμα.

**5. Τι καλείται διάγραμμα HIPO;**

Το διάγραμμα HIPO6 (Hierarchy Input Process Output = Ιεραρχία Εισόδου – Επεξεργασίας – Εξόδου) είναι μια τεχνική υλοποίησης του Δομημένου Προγραμματισμού και των αρχών του Ιεραρχικού και του Τμηματικού προγραμματισμού με τη χρήση **διαγραμμάτων**. Ένα διάγραμμα HIPO αποτελείται από **τρία βασικά τμήματα**, τα οποία περιγράφουν την είσοδο των δεδομένων, την επεξεργασία και τα αποτελέσματα που απαιτεί ένας αλγόριθμος και το αντίστοιχο πρόγραμμα. Στα διαγράμματα αυτά εμφανίζονται τα διάφορα **επίπεδα ιεραρχίας** και τα τμήματα του αλγορίθμου – προγράμματος με **αριθμό σύμφωνα με τη σειρά εκτέλεσής τους**. Στο ανώτερο επίπεδο υπάρχει μια περιγραφή της όλης διαδικασίας, στο αμέσως επόμενο επίπεδο διακρίνονται σε τμήματα οι τρεις βασικές λειτουργίες (είσοδος, επεξεργασία, έξοδος), οι οποίες μπορεί να αναλύονται περαιτέρω σε άλλα υπομήματα.

**6. Ποια είναι τα τρία βασικά τμήματα ενός διαγράμματος HIPO;**

Τα τρία βασικά τμήματα ενός διαγράμματος HIPO είναι αυτά που περιγράφουν την είσοδο των δεδομένων, την επεξεργασία και τα αποτελέσματα που απαιτεί ένας αλγόριθμος.

## 7. Ποιες είναι οι βασικές αρχές που χρησιμοποιεί ο δομημένος προγραμματισμός;

Ο δομημένος προγραμματισμός χρησιμοποιεί τις βασικές αρχές του **ιεραρχικού** και του **τμηματικού προγραμματισμού**.

## 8. Να συμπληρώσετε τα κενά με τη λέξη που λείπει:

α. Ο δομημένος προγραμματισμός προϋποθέτει την **ανάπτυξη** του αλγορίθμου.

β. Ο δομημένος προγραμματισμός αποτελείται από **ανεξάρτητα** μέρη με βάση ένα προκαθορισμένο σχέδιο.

γ. Ένα δομημένο πρόγραμμα είναι πολύ **πιο εύκολο** στην κατανόηση από προγράμματα που είναι γραμμένα με διαφορετικό τρόπο.

δ. Ένα δομημένο πρόγραμμα διαβάζεται εύκολα και επειδή αποτελείται από σαφώς καθορισμένα και **ανεξάρτητα** τμήματα, έχει εύκολη και τη **συντήρησή** του.

ε. Οι βασικές αλγοριθμικές δομές δεν είναι αρκετές για την κωδικοποίηση ενός **δομημένου** αλγορίθμου, αλλά πρέπει να αναπτυχθούν χρησιμοποιώντας συγκεκριμένες **τεχνικές**.

στ. Ένα διάγραμμα ΗΙΡΟ αποτελείται από τρία βασικά τμήματα, τα οποία περιγράφουν την **είσοδο** των δεδομένων, την **επεξεργασία** και τα **αποτελέσματα (έξοδο)**

6 Η απόδοση του ΗΙΡΟ (Hierarchical Input Output Processing) που δίνεται στις σελίδες 39 και 41 χρειάζεται διόρθωση σε (Hierarchy plus Input Process Output).

## Κεφάλαιο 6

### ΕΛΕΓΧΟΣ ΑΛΓΟΡΙΘΜΩΝ

#### Ερωτήσεις

#### 1. Τι καλείται δοκιμή αλγορίθμου;

Δοκιμή αλγορίθμου λέμε τον έλεγχο του αλγορίθμου για την ορθότητα των αποτελεσμάτων του

#### 2. Ποια είναι τα απαραίτητα βήματα για τον έλεγχο ενός αλγορίθμου;

Για να ελέγξουμε έναν αλγόριθμο εκτελούμε τα παρακάτω βήματα:

- Καταγράφουμε **εικονικά δεδομένα** σύμφωνα με τις απαιτήσεις του προβλήματος.
- Εξάγουμε τα αποτελέσματα σύμφωνα με τα βήματα του αλγορίθμου με **χειρογραφικό** τρόπο.
- **Κωδικοποιούμε** τον αλγόριθμο σε μία γλώσσα προγραμματισμού.
- Εκτελούμε το πρόγραμμα **σε υπολογιστή** και καταγράφουμε τα αποτελέσματα.
- **Συγκρίνουμε** τα χειρογραφικά και τα αποτελέσματα από την εκτέλεση του προγράμματος.

#### 3. Τι καλείται εκσφαλμάτωση;

Εκσφαλμάτωση (debugging) είναι η διαδικασία εύρεσης των λογικών λαθών ενός αλγορίθμου.

#### 4. Ποια η μεθοδολογία ανεύρεσης λογικών λαθών;

- Διαπίστωση του **είδους** του λάθους.
- Ανεύρεση του **ανεξάρτητου τμήματος** του αλγορίθμου που εκτελεί τη λανθασμένη λειτουργία.
- **Ανεύρεση του λάθους** μέσα σε αυτό το ανεξάρτητο τμήμα.

#### 5. Τι καλείται έλεγχος δεδομένων;

Έλεγχος δεδομένων καλείται η διαδικασία επιλογής δεδομένων με τέτοιο τρόπο έτσι ώστε αυτά να αντιστοιχούν και σε ένα διαφορετικό παράδειγμα καλύπτοντας κατ' αυτό τον τρόπο κάθε δυνατή περίπτωση πραγματικών δεδομένων.

#### 6. Τι καλείται αξιολόγηση αλγορίθμου;

Αξιολόγηση ενός αλγορίθμου ονομάζεται η διαδικασία της αποτίμησής του λαμβάνοντας υπόψη συγκεκριμένες παραμέτρους.

#### 7. Ποια τα βασικά κριτήρια αξιολόγησης;

Τα βασικά κριτήρια αξιολόγησης ενός αλγορίθμου είναι:

- Η δυνατότητα που παρέχει για ελαχιστοποίηση του κόστους ελέγχου της ορθότητάς του.
- Η δυνατότητα που παρέχει για εύκολη συντήρηση και ευκολία για αλλαγές και επεκτάσεις.
- Η δυνατότητα που παρέχει για εύκολη ανάπτυξη σε γλώσσα προγραμματισμού, ανάλογα με την υφή του.

#### 8. Για ποιο λόγο σήμερα μπορούμε να έχουμε ευχερέστερο έλεγχο της ορθότητας ενός αλγορίθμου;

Σήμερα, ο έλεγχος της ορθότητας ενός αλγορίθμου είναι ευχερής διότι α) είναι **προσιτή η χρήση του υπολογιστή** (από άποψη τόσο του υλικού όσο και του λογισμικού) σε αντίθεση με τα παλαιότερα χρόνια και β) οι γλώσσες προγραμματισμού έχουν αναπτυχθεί πάρα πολύ, ώστε να είναι **εύχρηστες** τόσο, όσο ο ψευδοκώδικας.

#### 9. Να συμπληρώσετε τα κενά με τις λέξεις που λείπουν:

α. Μετά την αποτύπωση του αλγορίθμου ενός προβλήματος ακολουθεί η διαδικασία της **δοκιμής** και της **αξιολόγησης** του προγράμματος.

- β. Για τον έλεγχο του αλγορίθμου που σχεδιάσαμε, πρέπει να δημιουργήσουμε **δεδομένα** και με χειρογραφικό τρόπο να εξαγάγουμε τα αποτελέσματα σύμφωνα με τον αλγόριθμο.
- γ. Σήμερα μπορούμε με **ευκολία** να δοκιμάζουμε την ορθότητα ενός αλγορίθμου, αφού τον κωδικοποιήσουμε σε γλώσσα προγραμματισμού.
- δ. Πρωταρχικός παράγων επιτυχίας της δοκιμής του αλγορίθμου μας είναι η του προβλήματος.
- ε. Η αξιολόγηση ενός αλγορίθμου εξαρτάται άμεσα και από τη **γλώσσα προγραμματισμού** στην οποία θα υλοποιηθεί.
- στ. Η αξιολόγηση βέβαια ενός αλγορίθμου **εξαρτάται** και από την επίδοσή του σε χρόνο εκτέλεσης του προγράμματος.
- ζ. Η αξιολόγηση ενός αλγορίθμου εξαρτάται και από την απασχόληση της **μνήμης** ή άλλων αντίστοιχων **πόρων** του συστήματος.

## Κεφάλαιο 7

### ΕΙΔΗ, ΤΕΧΝΙΚΕΣ ΚΑΙ ΠΕΡΙΒΑΛΛΟΝΤΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

#### 1. Να αναφέρετε διαφορές μεταξύ γλωσσών μηχανής και γλωσσών χαμηλού επιπέδου.

- Οι γλώσσες μηχανής κωδικοποιούν τις εντολές τους **με ομάδες 0 και 1** σταθερού ή μεταβλητού πλάτους ενώ οι γλώσσες χαμηλού επιπέδου με συντομογραφίες **λέξεων** της αγγλικής γλώσσας.
- Ένα πρόγραμμα σε γλώσσα μηχανής εκτελείται **απευθείας** από την ΚΜΕ του υπολογιστή ενώ ένα πρόγραμμα γλώσσας χαμηλού επιπέδου απαιτεί **μεταφραστικό πρόγραμμα** για να μετατραπεί σε γλώσσα μηχανής.

#### 2. Να αναφέρετε διαφορές μεταξύ γλωσσών υψηλού επιπέδου και γλωσσών χαμηλού επιπέδου.

- Ένα πρόγραμμα σε γλώσσα υψηλού επιπέδου δεν είναι εξαρτημένο από τον υπολογιστή ενώ ένα πρόγραμμα σε γλώσσα χαμηλού επιπέδου είναι.
- Μια γλώσσα υψηλού επιπέδου βασίζεται σε λέξεις κλειδιά της αγγλικής γλώσσας ενώ μια γλώσσα χαμηλού επιπέδου όχι.
- Μια εντολή σε γλώσσα υψηλού επιπέδου μεταφράζεται σε πολλές εντολές του κώδικα μηχανής κατά τη διάρκεια της μετάφρασης ενώ μια εντολή σε γλώσσα χαμηλού επιπέδου μεταφράζεται σε μία μόνο εντολή γλώσσας μηχανής.

#### 3. Πώς μπορούμε να χαρακτηρίσουμε τις γλώσσες υψηλού επιπέδου ανάλογα με τις λειτουργικές τους ιδιότητες;

Σε γλώσσες για

- α: Επιστημονικές εφαρμογές (Fortran, C, APL).  
 β: Εμπορικές εφαρμογές (Cobol, RPG).  
 γ: Εκπαιδευτικές εφαρμογές (Basic, Logo).  
 δ: Ειδικές εφαρμογές (Lisp, Prolog).  
 ε: Επιστημονικές – εμπορικές εφαρμογές (PL-1- PASCAL).

#### 4. Τι γνωρίζετε για τις γλώσσες 4ης γενιάς;

- Είναι γλώσσες **υψηλής παραγωγικότητας** και μπορούν να τις χρησιμοποιήσουν και άτομα που δεν έχουν γνώσεις προγραμματισμού.
- Διαθέτουν **δυναμικές εντολές** για τη δημιουργία αρχείων, τη δημιουργία και ενημέρωση οθονών για την εκμετάλλευση των αρχείων και τη δημιουργία εκτυπώσεων. Κάθε μία εντολή γλώσσας 4ης γενιάς αντιστοιχεί σε ολόκληρο πρόγραμμα μιας κλασσικής γλώσσας προγραμματισμού.
- Είναι γλώσσες **διαλογικές**.
- Η **υλοποίηση** πληροφοριακών συστημάτων είναι ταχύτερη με χρήση γλωσσών 4ης γενιάς.

#### 5. Τι γνωρίζετε για τον διαδικασιακό προγραμματισμό;

Στο Διαδικασιακό προγραμματισμό, το πρόγραμμα είναι γραμμένο σε κάποια γλώσσα εξαρτημένη από τη διαδικασία, η οποία είναι χαρακτηριστικό των γλωσσών 3ης γενιάς. (COBOL, FORTRAN, PASCAL). Μια γλώσσα που είναι εξαρτημένη από τη διαδικασία **δίνει έμφαση στον προγραμματισμό των υπολογιστικών και λογικών διαδικασιών που απαιτούνται για την επίλυση του προβλήματος**.

Προβλήματα που επιλύονται με διαδικασιακό προγραμματισμό είναι για λογιστικές, εμπορικές, διοικητικές εφαρμογές, εφαρμογές μεταφορών κτλ.

Στο διαδικασιακό προγραμματισμό, το **πρόγραμμα** είναι μια διαδικασία η οποία εκτελεί **ένα προς ένα τα βήματα του αλγόριθμου επίλυσης του προβλήματος**. Οι εντολές εκτελούνται **διαδοχικά** εκτός αν υπάρχουν **συνθήκες** οπότε κάποιες εντολές παραλείπονται και δημιουργούνται **διακλαδώσεις** ή κάποιες εντολές **επαναλαμβάνονται**.

Ο διαδικασιακός προγραμματισμός, με τις γλώσσες 3ης γενιάς έχει απομακρύνει τον προγραμματισμό από το επίπεδο της μηχανής και τον έχει φέρει προς το **επίπεδο του ανθρώπου**.

#### 6. Να δώσετε μερικά παραδείγματα γλωσσών που είναι εξαρτημένες από τη διαδικασία.

COBOL, FORTRAN, PASCAL και επίσης οι Ada, Modula 2, Concurrent C.

**7. Ποιες είναι οι βασικές αλγοριθμικές δομές και ποιες οι τεχνικές σχεδίασης στο δομημένο προγραμματισμό;**

Στον δομημένο προγραμματισμό, οι βασικές αλγοριθμικές δομές είναι οι δομές της **διαδοχής**, της απλής **επιλογής** και της **επανάληψης**.

Στον δομημένο προγραμματισμό χρησιμοποιούνται οι τεχνικές του **ιεραρχικού** και του **τμηματικού** προγραμματισμού.

**8. Πού χρησιμοποιούνται τα διαγράμματα HIPO;**

Τα διαγράμματα HIPO χρησιμοποιούνται στον δομημένο προγραμματισμό για την αναπαράσταση των τεχνικών του ιεραρχικού σχεδιασμού και του τμηματικού προγραμματισμού. Στα διαγράμματα αυτά εμφανίζονται τα διάφορα επίπεδα ιεραρχίας και τα διάφορα τμήματα προγράμματος με αριθμό σύμφωνα με τη σειρά εκτέλεσής τους.

**9. Πού οφείλει την καθιέρωσή του ο παράλληλος προγραμματισμός;**

Ο Παράλληλος Προγραμματισμός οφείλει την καθιέρωσή του στην **ανάπτυξη του υλικού**. Η εμφάνιση της αρχιτεκτονικής των **πολλών επεξεργαστών** οι οποίοι χρησιμοποιούν **κοινή μνήμη** είχε ως συνέπεια την ανάπτυξη των παράλληλων αλγορίθμων οι οποίοι και επέβαλαν τον Παράλληλο Προγραμματισμό. Αντιπροσωπευτική γλώσσα για παράλληλη επεξεργασία είναι η γλώσσα Occam.

**10. Τι είναι ο αντικειμενοστρεφής προγραμματισμός;**

Ο αντικειμενοστρεφής προγραμματισμός είναι μία τεχνική στην οποία υπάρχει ενσωμάτωση των δεδομένων και του τρόπου χειρισμού τους μέσα από την έννοια του **αντικειμένου**. Τα **χαρακτηριστικά** ενός αντικειμένου είναι τα **δεδομένα από τα οποία αποτελείται** και οι **μέθοδοι** του αντικειμένου είναι οι ενέργειες οι οποίες σχετίζονται με την επεξεργασία των δεδομένων και **καθορίζουν τη συμπεριφορά του αντικειμένου** στο πρόγραμμα. Οι μέθοδοι μπορεί να είναι **διαδικασίες** ή **συναρτήσεις** του χρήστη στο κυρίως πρόγραμμα. Σε ένα αντικειμενοστρεφές πρόγραμμα, μπορούν να δημιουργούνται **κλάσεις ιεραρχικά δομημένες**. Με την ιεραρχική δόμηση των κλάσεων, οι **υποκλάσεις κληρονομούν τα χαρακτηριστικά και τις ιδιότητες των κλάσεων** από τις οποίες προήλθαν, έχοντας τη δυνατότητα να προστεθούν σε αυτές (δηλ. στις υποκλάσεις) **νέες ιδιότητες** και χαρακτηριστικά.

**11. Ποια είναι τα χαρακτηριστικά και ποιες οι μέθοδοι στον αντικειμενοστρεφή προγραμματισμό;**

Τα **χαρακτηριστικά** ενός αντικειμένου είναι τα **δεδομένα από τα οποία αποτελείται** και οι **μέθοδοι** του αντικειμένου είναι οι ενέργειες οι οποίες σχετίζονται με την επεξεργασία των δεδομένων και **καθορίζουν τη συμπεριφορά του αντικειμένου** στο πρόγραμμα. Οι μέθοδοι μπορεί να είναι **διαδικασίες** ή **συναρτήσεις** του χρήστη στο κυρίως πρόγραμμα.

**12. Τι είναι ο Συναρτησιακός Προγραμματισμός;**

Ο Συναρτησιακός Προγραμματισμός **στηρίζεται στην έννοια της συνάρτησης**, όπως την έχουμε γνωρίσει από τα μαθηματικά. Η συνάρτηση δίνει τη δυνατότητα απεικόνισης ενός πεδίου ορισμού σε ένα πεδίο τιμών. Στο Συναρτησιακό προγραμματισμό οι **εντολές** και οι **δομές ελέγχου είναι συναρτήσεις** οι οποίες έχουν ως ορίσματα δεδομένα ή άλλες συναρτήσεις.

**13. Αναφέρετε παραδείγματα γλωσσών Συναρτησιακού Προγραμματισμού.**

Παραδείγματα γλωσσών Συναρτησιακού Προγραμματισμού είναι οι γλώσσες FORTH και LOGO.

**14. Ποια η διαφορά του Λογικού από το Διαδικασιακό προγραμματισμό;**

Στον Λογικό προγραμματισμό, στο πρόγραμμα **δεν περιγράφεται ο αλγόριθμος** επίλυσης του προβλήματος αλλά **η γνώση** που σχετίζεται με το πρόβλημα. Ο **Λογικός Προγραμματισμός σε αντίθεση με το Διαδικασιακό, στηρίζεται στη γνώση**. Η γνώση περιγράφεται υπό μορφή **γεγονότων** και **κανόνων** ενώ **ορίζεται μία πρόταση – στόχος προς απόδειξη**. Για την απόδειξη της πρότασης αυτής το πρόγραμμα εφαρμόζει τους κανόνες του πάνω στα γεγονότα.

**15. Να εξηγήσετε τους όρους πηγαίο πρόγραμμα, αντικείμενο πρόγραμμα και μεταγλωττιστής.**

**Πηγαίο πρόγραμμα** (source) ονομάζεται το πρόγραμμα το οποίο είναι γραμμένο σε μια γλώσσα προγραμματισμού<sup>9</sup>. Ο **μεταγλωττιστής** είναι ένα ειδικό λογισμικό (πρόγραμμα) το οποίο μετατρέπει το πηγαίο πρόγραμμα σε **αντικείμενο** (object) πρόγραμμα. Το αντικείμενο πρόγραμμα περνά στη συνέχεια από κάποιες διεργασίες δίνοντας το πρόγραμμα σε κώδικα (**γλώσσα**) **μηχανής** το οποίο μπορεί άμεσα να εκτελέσει ο υπολογιστής<sup>10</sup>.

**16. Να αναφέρετε ομοιότητες και διαφορές μεταξύ compiler και interpreter.**

Ομοιότητες:

- Τόσο ο compiler όσο και ο interpreter είναι μεταφραστικά προγράμματα τα οποία μετατρέπουν ένα πρόγραμμα από γλώσσα προγραμματισμού<sup>11</sup> σε γλώσσα μηχανής ώστε να μπορεί να εκτελεστεί από τον υπολογιστή.

<sup>11</sup> Υψηλού επιπέδου <sup>12</sup>Το πηγαίο πρόγραμμα πρέπει να υπάρχει στον υπολογιστή και να εκτελείται σε συνδυασμό με τον interpreter. <sup>13</sup> compiler

Διαφορές:

- Ο compiler κάνει ανάλυση(συντακτική και γραμματική) ολόκληρου του πηγαίου προγράμματος και το μετατρέπει σε αντικείμενο πρόγραμμα μόνο εφόσον δεν υπάρχουν λάθη. Αντίθετα, ο interpreter δεν ελέγχει τη σύνταξη ολόκληρου του πηγαίου προγράμματος πριν από την εκτέλεση.
- Με τη χρήση compiler και εφόσον γίνει επιτυχής μετάφραση, το αντικείμενο πρόγραμμα μπορεί να χρησιμοποιηθεί όσες φορές θέλουμε χωρίς να προηγηθεί μετάφραση. Αντίθετα, με τη χρήση interpreter, πάντα θα πρέπει να επαναλαμβάνεται η διαδικασία μετάφρασης – ελέγχου κάθε φορά που εκτελείται το πηγαίο πρόγραμμα<sup>12</sup>. Ο συντακτικός έλεγχος και η μετάφραση γίνεται σε κάθε μία από τις εντολές του πηγαίου προγράμματος . Εφόσον δεν υπάρχουν λάθη, παράγεται η αντίστοιχη εντολή σε κώδικα μηχανής, εκτελείται και συνεχίζεται η μετάφραση της επόμενης εντολής.
- Ο έλεγχος και η διόρθωση των λαθών ενός προγράμματος με τη χρήση interpreter είναι πιο εύκολη από ότι η διόρθωση με τη χρήση compiler, αφού στην περίπτωση του interpreter το πρόγραμμα σταματά στο σημείο του λάθους, ο προγραμματιστής το πληροφορείται, το διορθώνει και συνεχίζει την εκτέλεση του προγράμματος.
- Η εκτέλεση ενός προγράμματος με interpreter είναι πολύ πιο αργή σε σύγκριση με την εκτέλεση του ίδιου προγράμματος, μεταφρασμένου με compiler.

### 17. Να εξηγήσετε τη διαδικασία γραφής –εκτέλεσης προγράμματος.

Το **πηγαίο πρόγραμμα** γράφεται σε κάποια γλώσσα προγραμματισμού με τη βοήθεια ενός απλού προγράμματος επεξεργασίας κειμένου (που λέγεται **εκδότης – διορθωτής (editor)**).

Ο **μεταγλωττιστής**<sup>13</sup> ελέγχει το πηγαίο πρόγραμμα για ορθογραφικά ή συντακτικά λάθη.

Σε περίπτωση που διαπιστώνονται λάθη, αυτά διορθώνονται και το πηγαίο πρόγραμμα επαναλέγχεται από το μεταγλωττιστή. Η διαδικασία αυτή επαναλαμβάνεται μέχρι τα λάθη να μηδενιστούν.

Στην περίπτωση αυτή, η μεταγλώττιση του προγράμματος είναι επιτυχής και παράγεται το **αντικείμενο πρόγραμμα**. Το αντικείμενο πρόγραμμα δεν είναι κατ' ανάγκη εκτελέσιμο από τον υπολογιστή αλλά πρέπει να **συνδεθεί** με **υποπρογράμματα της βιβλιοθήκης της γλώσσας** ή του προγραμματιστή με τη βοήθεια ενός άλλου προγράμματος του **συνδέτη (linker)** ώστε τελικά να παραχθεί εκτελέσιμος κώδικας.

[διαδικασία: **γραφή** → **μεταγλώττιση** → **σύνδεση** → **εκτέλεση**].

Σε άλλες περιπτώσεις αντί του συνδέτη χρησιμοποιείται ένα άλλο πρόγραμμα, ο **φορτωτής (loader)** ο οποίος συνδέει το αντικείμενο πρόγραμμα με υποπρογράμματα της βιβλιοθήκης της γλώσσας ή του προγραμματιστή και φορτώνει στη μνήμη για εκτέλεση του κώδικα μηχανής που προκύπτει. [διαδικασία: **γραφή** → **μεταγλώττιση** → **φόρτωση**].

Στην περίπτωση που το μεταφραστικό πρόγραμμα είναι διεργασία, η μεταγλώττιση, η σύνδεση και η εκτέλεση γίνονται εντολή προς εντολή κατά τη σειρά εκτέλεσής τους.

### 18. Ποια είναι τα σύγχρονα προγραμματιστικά περιβάλλοντα;

Τα σύγχρονα προγραμματιστικά περιβάλλοντα εμπεριέχουν σε ένα **ολοκληρωμένο περιβάλλον**, πολλά από τα προγράμματα ανάπτυξης, όπως διορθωτή (editor), μεταγλωττιστή (compiler), εκσφαλματωτή (debugger) και άλλα στα οποία μπορούμε να έχουμε πρόσβαση μέσα από κατάλληλα μενού. Επιπλέον, ένα ολοκληρωμένο περιβάλλον ανάπτυξης μπορεί να διαθέτει και εργαλεία σχεδίασης με τη βοήθεια γραφικών οπότε χαρακτηρίζεται ως **γραφικό περιβάλλον**.

### 19. Να συμπληρώσετε τα κενά από τη λέξη που λείπει:

- ανάπτυξης προγράμματος
- κωδικοποίησή
- ακολουθία , μεταβλητού
- συντομογραφίες, θέσεις , συμβολικές χαμηλού
- επιπέδου, αγγλικής, γραμματική, συντακτικό, προτάσεις, εντολές
- 4ης, εκτέλεσής, διαλογικές, εντολών

### 20. Να συμπληρώσετε τα κενά με τη λέξη που λείπει:

- διαδικασία, 3ης γενιάς
- αλγοριθμικές, διαδοχή, επιλογή, επανάληψη, ιεραρχικού, τμηματικού, HIPO, Ιεράρχηση, επεξεργασίας.
- πολλών, κοινή, παράλληλων, προγραμματισμού.
- χειρισμού, αντικειμένου, χαρακτηριστικά, επεξεργασία, αντικειμένου, διαδικασίες.
- μεταφράζει, προγραμματισμού, μηχανής, πηγαίο, αντικείμενο.